加微信：642945106 发送“ 赠送 ”领取赠送精品课程
发数字“ 2 ”获取众筹列表

02 | 系统可用性：没有故障，系统就一定是稳定的吗？

2020-03-18 赵成

SRE实战手册 进入课程 >

讲述：赵成
时长 11:43　大小 10.73M

你好，我是赵成，欢迎回来。

我们先来复习一下上一讲的内容，总结下来就是，SRE 是个体系化工程，我们通过构建 SRE 这样一套体系来保证系统稳定性，具体来说就是“提升 MTBF，降低 MTTR”。有了这样一个激动人心的目标，你是不是想着那咱还等什么，赶快、立马就入手建设 SRE 体系吧！

嗯，好想法，我也很想咱就直接“撸起袖子加油干”。不过今天我们要先缓一缓，在正式入 SRE 落地细节之前，我们得先讨论一下目前业界常用的“系统可用性（Availability）”这个概念，也就是我们常常听到的“3 个 9”（99.9% 或 99.95%）、“4 个 9”（99.99% 或 99.995%）。

为什么要先来讨论“系统可用性”这个大家已经很熟悉的概念呢？

一方面，系统可用性和我们建设 SRE 的目标强相关，SRE 的稳定性目标其实就是尽量减少系统故障或异常运行状态的发生，提升系统可用的运行时间占比。很明显，这个可用时长就非常关键了。

另一方面，系统可用性这个概念看似简单，但我发现真的深入进去，大家的理解其实有很多不一致的地方，比如到底怎样才算是可用时长，怎样算是不可用时长呢？这个标准是怎么定义的？除了从时间维度来衡量可用性，还有其它的衡量方式吗？“3 个 9”、“4 个 9”听起来都很好，那具体来说我们的系统要达到“几个 9”才算是稳定的呢？

所以，今天我们先慢下来，花时间把上面这些问题都彻底搞清楚，达成共识，打好基础，咱后面的 SRE 学习才能事半功倍。

衡量系统可用性的 2 种方式

那我就直接给答案了，目前业界有两种衡量系统可用性的方式，一个是时间维度，一个是请求维度，我们先来看这两个维度的计算公式。

时间维度：Availability = Uptime / (Uptime + Downtime)

请求维度：Availability = Successful request / Total request

这两个公式很简单，我们得深入进去，一一来看。

我们先来看时间维度的系统可用性。用一句话来概括：**时长维度，是从故障角度出发对系统稳定性进行评估**。

这类计算方式我们最常见，毕竟你的系统在一段时间里不出现故障，就说明它很稳定嘛！不过，在真实的使用场景中，怎么样才算是可用时长，什么情况下又是不可用时长，这个是怎么定义的呢？

细想一下这个问题，你会发现还真有点复杂，那我就举个发烧生病的例子来说明一下。

我们知道，一个人如果发烧了，体温一般会超过 37.5 度，那如果这个人的体温正好达到这个温度，是不是代表他一定是生病了呢？依据生活经验，我们知道不一定。为什么呢？因为我们判断一个人是否发烧生病，不是只看这一次、一时的体温，还要看他体温是不是持续超过 37.5 度。

所以，这里就涉及到一个测量方法和判定方法的问题，包含三个要素：一个是**衡量指标**，比如体温就是衡量指标；第二个是**衡量目标**，达到什么目标是正常，达不到就是异常，低于 37.5 度算正常，超过 37.5 度就是异常，但是单次测量不能说明问题，我们可以多次测量，比如 6 次中有至少 4 次低于 37.5 度才算正常，转化成比例的话就是 67%；第三个是**影响时长**，比如持续超过 12 小时。

对应到系统上，我们也会用一系列的标准和判定逻辑来说明系统是否正常。比如，系统请求状态码为非 5xx 的比例，也就是请求成功率低于 95%，已经连续超过 10 分钟，这时就要算作故障，那么 10 分钟就要纳入 Downtime（宕机时间），如果达不到这个标准，就不算作故障，只是算作一般或偶然的异常问题。

这里同样有三个要素：**衡量指标**，系统请求状态码；**衡量目标**，非 5xx 占比，也就是成功率达到 95%；**影响时长**，持续 10 分钟。

因此，只有当问题达到一定影响程度才会算作故障，这时才会计算不可用时长，也就是上面公式中的 Downtime。同时，我们还要求一个周期内，允许的 Downtime，或者说是系统的“生病时间”是有限的，用这个有限时间来约束系统稳定性。

下面是我们常见的按时长维度统计的可用性对照表，也就是我们前面提到的几个 9：

防止断更 请务必加
首发微信：17161143665

可用性对照表

极客时间

系统可用性	故障时间/年	故障时间/月	故障时间/周	故障时间/天
90%	36.5天	72小时	16.8小时	2.4小时
99%	3.65天	7.2小时	1.68小时	14.4分
99.9%	8.76小时	43.8分钟	10.1分钟	1.44分
99.99%	52.56分钟	4.38分钟	1.01分钟	8.66秒
99.999%	5.26分钟	25.9秒	6.05秒	0.87秒

讲到这里，针对时长维度的稳定性计算方式就比较清楚了，但是从这种计算方式中，你有没有看出一些问题呢？

我想你肯定看出来了，这里最显著的问题就是，稳定性只与故障发生挂钩。

我们来想一想，这样做会带来哪些问题？比如有一个系统，因为网络抖动，有短暂的几秒、十几秒，或者几分钟异常，但是后来系统自己恢复了，业务并没有中断，这时我们按照时长维度来判断，这肯定不会算作系统故障。但是如果这种短暂的影响频度非常高，一天来个 5、6 次，持续一两周，我们应该可以判定系统运行状况也是不正常的，可能不是故障，但肯定是不稳定了。

所以这种用时长维度来衡量系统稳定性的方式，其主要缺点就是粒度不够精细。这些小的异常问题和它们的影响，如果从更长的周期来看，也是有一定参考价值的。那怎样才能衡量得更精细些呢？

这就需要第二种衡量方式了，也就是从请求维度来衡量系统可用性。

用一句话来说，**请求维度，是从成功请求占比的角度出发，对系统的稳定性进行评估。**

假定我们的系统一天内有 100,000 次请求，我们期望的成功率至少是 95%，如果有 5001 次请求失败了，也就是成功率低于 95% 了，我们就认为系统运行状态是不正常的。

请求维度的系统可用性同样包含三个关键要素，第一个**衡量指标**，请求成功率；第二个**衡量目标**，成功率达到 95% 才算系统运行正常；第三个是**统计周期**，比如一天、一周、一个月等等，我们是在一个统计周期内计算整体状况，而不是看单次的。

你看，这种方式对系统运行状况是否稳定监管得更为严格，不会漏掉任何一次问题的影响，因为它对系统整体运行的稳定性判定，不仅仅会通过单次的异常影响进行评估，还会累计叠加进行周期性的评估。

到这里，我们就总结出一条至关重要的经验了：**故障一定意味着不稳定，但是不稳定，并不意味着一定有故障发生。**

到这里，我们掌握了衡量系统可用性的两个维度、两种算法，它们都包含三个关键要素：衡量指标、衡量目标、影响时长 / 统计周期。这两种算法最后都会落脚到“几个 9”上，那系统到底定“几个 9”才算是稳定的呢？接下来，我们就来回答这个问题。

拼课微信：17161436665

设定系统稳定性目标要考虑的 3 个因素

这个问题其实并没有标准答案，从我的经验来看，到底定“几个 9”主要取决于以下三个因素。

第一个，成本因素。

从理论上来说，肯定是 9 越多稳定性越好，但是相应付出的成本和代价也会更高。比如为了更高的可用性，要有更多的冗余资源投入，甚至要做主备、双活甚至是多活。如果一家公司的业务量和影响力都发展到一定程度，那这个成本不管多高都是必须要付出的。但是，肯定不是所有的公司都需要付出这么高的成本，而是要先考虑 ROI（回报率）。这时候就要看企业自身对成本压力的承担情况了。

第二个，业务容忍度。

稳定性怎么设定，很大程度上还要取决于业务上的容忍度。对于核心业务或核心应用，比如电商的交易和支付系统，我们当然是希望成功率越高越好，一般对系统稳定性要求是“3 个

9”或“4 个 9”。因为这些系统一旦出问题，就会直接影响整个网站和公司的收益，这些都是钱，所以对稳定性要求必然就会提高。

但是，对于非核心业务或应用，比如商品评论，商品评分等，或许“2 个 9”也能容忍。因为短时间的评论看不到，并不会对业务收入和用户体验造成太大的影响。

第三个，系统当前的稳定性状况。

结合系统的实际情况，**定一个合理的标准比定一个更高的标准会更重要**。这个合理的值应该怎么来定呢？

我个人的建议是从系统现状入手，比如，如果系统可用性是低于 99% 的，那首先第一步是不是可以做到 99%，然后再争取做到 99.5%，再到 99.9%，一步一步朝着更高的标准迈进。同时，这样做也会更容易落地，因为你如果定一个太高的目标，又始终达不成，反而会打击到团队的自信心和积极性。

结合上面这三个因素，对于到底应该定“几个 9”这个问题，你应该有了一个更清晰的认识了。

总结

好了，到这里，今天我们要讨论的系统可用性就讲完了。关于系统可用性，业界有两种计算方式，一种是时长维度，另一种是请求维度，这两种方式各有优劣。在 SRE 的实践中，应该选择哪一个呢？很明显，SRE 会更多采用请求维度的统计方式，因为 **SRE 关注的稳定性是系统的整体运行状态，而不仅仅只关注故障状态下的稳定性，在系统运行过程中的任何异常，都会被纳入稳定性的评估范畴中。**

这个知识点要拿一整节课来讲，是因为接下来我们就要讨论 SRE 的稳定性指标和目标了，理解了今天的内容，你才能更好地理解 SRE 体系中的指标（SLI）和目标（SLO）。今天我先把 SLI 和 SLO 这两个概念抛出来，如果你觉得有点陌生，没有关系，准备好下节课和我一起掌握它们。

思考题

对于系统可用性的描述，今天我们仅用了“状态码”这一个指标来示例，但是在实际情况下，我们还会有其它多个指标来同时标识一个系统的稳定性，你能想到还有哪些指标？欢迎你在留言区写下自己的思考。

考虑这些指标的时候，不妨想想你是怎么选择的，你的判断标准是什么？这些也将是我们下节课程的重点内容。

如果今天的内容对你有帮助，也欢迎你分享给身边的朋友，和他一起精进。

我是赵成，我们下节课见。

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇　01 | SRE迷思：无所不能的角色？还是运维的升级？

下一篇　03 | SRE切入点：选择SLI，设定SLO

精选留言 (13)

写留言

春来草自青

2020-03-19

从业务部门的视角来看，状态码是多少他们是不关心的，他们关心的是业务是否真正的可用。比如，极端一点，状态码正常，但返回的内容不是预期的。

另外，如果业务不是需要7*24的，可用性指标应该是仅限定在业务开展期间。
...
展开

作者回复: 很好的问题。从两个方面来看：

1、返回的业务内容不是预期的，这一点应该更多的是要QA来解决，这个本质是是功能问题，其实SRE是不需要关注这样的异常的。

2、如果可以做到内容不同，返回码不同，也可以做到稳定性的监控。比如对于一个用户登录应用，如果登录成功是200ok，验证码错误是1001，密码错误是1002，如果总是提示验证码错误，这种也是可以纳入到稳定性的监控指标中的。

关于第二个问题，7*24小时值守，这个看业务特点，比如对于证券类的应用，每天就是几个固定时段，所以没必要7*24小时。

5

leslie
2020-03-18

就像老师课程中所提及的三要素”成本因素、业务容忍度、系统当前的稳定性状况“。这三点无一不需要综合考虑，甚至有时都得考虑三者之间的比例。
个人在此之外会考虑”离散度“：确实有时觉得好像还稳定，可是离散度是否正常。记得老师的推荐的书籍中就提及过，"系统正常是系统异常的特殊情况";有故障才是是常态。没有故障且稳定说明大家都在做机械化重复操作，如何从故障和不稳定中找出问题才是根...
展开

作者回复: 你的分享也非常用心，总能抓住很多关键点，很赞！

3

foxracle
2020-03-19

系统可用性从用户角度来评价最直接，至少包括
1：能否访问到。这应该包括从用户操作到收到响应整个端到端的是否可用，对于用户最后一公里的不可控段也需要能感知和预警。
2：访问是否顺畅。响应时间是否异常

展开

作者回复: 从用户角度来评价，这个点把握的非常准确。

1

小圣
2020-03-19

简洁，清晰

展开

作者回复: 谢谢！看看能不能用简介清晰的几句话表达出来呢^_^

1

张春阳
2020-03-22

可用性指标，我觉得还需要区分下
1. 业务可用性，可以称之为“源站”“ ，这里的统计指标最好给业务提供最大的灵活性，授权rd自身配置uei，请求方法，以及正常状态码。这里需要注意一个问题，有时候301/302/499也是错误码（比如为了友好提示5xx内部跳转，触发请求限流503），所以最好将body自定义内容作为判断指标。...

展开

作者回复: 感谢你提供了用户可用性这个维度，从SRE角度，离用户越近的指标，越有价值。

大尾巴老猫
2020-03-21

RED、USE、four golden signals

展开

作者回复: 很有意思的回答，我再提一个问题，就是four golden signals适合做衡量稳定性的指标吗?

大尾巴老猫
2020-03-21

请教一下，老师怎么看AIOPS，AIOPS对线上的业务来说，真的有（真实的）价值吗？

作者回复: 要清楚，AIOps对业务本身是没有价值的，它是为运维服务的，所以它的价值更多的体现在运维层面。

体现在运维层面的哪里呢？就是问题发生前的预判、以及根因分析这些，因为在大规模分布式系统下，没有这个手段就没法处理问题。

但是AI要有两个要素，一个是算法，一个是数据，算法是靠数据训练的，这里算法不是问题，但是数据量是不是足够大就是问题，所以一定要业务体量足够大，资源规模足够大，产生的日志信息足够丰富，这时AIOps才有意义，如果只有几十几百台服务器的规模，业务体量也没有那么大的情况下，AIOps的作用是不大的。

sda
2020-03-21

系统的稳定性的衡量指标最关键要与业务结合一起。
我认为最重要的是要引入服务的健康检查机制，具体说明：
如A服务依赖诸多中间件例如数据库、消息队列等，也依赖其他B、C等服务。A服务对检查的服务所依赖的不同的中间件或服务定义为HealthCheck的最终结果为true或false，任何一个HealthCheck的healthy结果为false，那么最终状态就会是false。{...

展开

作者回复: 你提供了一个很好的案例。你提到的是针对单次请求的成功与否，但是如果要判断服务是不是正常，单次判断是不是够？如果不够，那应该采取什么样的判断方式呢？

可以尝试思考下这两个问题。

1

Helios
2020-03-21

这一节感觉类似于监控～
可收集的指标有
机器性能，比如说磁盘io，cpu占用率，内存使用率，这些在空闲的时候和忙碌的时候对请求的影响肯定不一样，如果是对读写有要求的应用，磁盘读写会影响很大。
...

展开

作者回复: 这个问题你可以先思考下，因为接下来的一讲，我们就会详细讲解这部分内容。

海华(海菜)
2020-03-20

可用性和新功能是矛盾，考虑到这个矛盾关系，可用性目标还跟新功能的需求有关系吧，例如公司/产品快速增长阶段甚至能适当容忍可用性故障

展开

作者回复: 是的，就是业务容忍度。

不过你提到的在快速增长阶段要特殊考虑，确实也是一个需要考虑的维度和因素。

云峰
2020-03-20

请求成功率的判定需要一定分类和标准。请求可以分类，静态请求，动态请求。动态请求还可以按业务模块进行细分，这样故障时初步就可以定位故障位置，影响范围。静态请求比较好判定，排除网络问题，基本可以访问到就可以。动态请求，这个得具体分析，有的请求比较复杂，正常就慢，这块判断就得靠人员经验和历史数据了。

展开

作者回复: 最后一句，动态请求要靠经验和历史数据，其实也可以形成一种度量标准，接下来的第03部分，就会介绍到。

OlafOO
2020-03-20

机器性能指标
应用层级指标
服务质量指标

展开

作者回复: 赞!

蓝魔丶

2020-03-20

响应时间，吞吐量，系统资源占用

展开

作者回复: 非常好，如果能尝试再分个类就更棒了。再就是，要看下，响应时间、吞吐量、资源占用，他们的主体是谁?